Πανεπιστήμιο Πειραιώς

Σχολή Τεχνολογιών Πληροφορικής και Τηλεπικοινωνίων

Τμήμα Ψηφιακών Συστημάτων

ΤΙΤΛΟΣ ΕΡΓΑΣΙΑΣ :

Απαλλακτική εργασία μαθήματος (Εαρ. Εξ. 2020-21)

ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ :

❖ ΤΣΟΥΦΗΣ ΘΕΟΔΩΡΟΣ
❖ ΑΜ: Ε17155
❖ Εξάμηνο σπουδών: 8

Εργασία στο μάθημα «Διακυβέρνηση Πληροφοριακών Συστημάτων»

Επιβλέπουσα: κ.Μαυρογιώργου Αργυρώ.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΝΟΤΗΤΑ 1: Θεματολογία Εφαρμογής Ανάλυσης Δεδομένων

Η εφαρμογή αυτή μαζεύει live streaming δεδομένα από το twitter . Τα δεδομένα που μαζεύει (όταν λέμε δεδομένα εννοούμε tweets χρηστών) περιέχουν κάποιες από τις ακόλουθες λέξεις κλειδιά → covid-19 , vaccine , vaccination , vaccinated , AstraZeneca , Pfizer .

Σκοπός της εφαρμογής είναι από τα live streaming δεδομένα που θα συλλέξει (τα οποία είναι 2000) , να κάνει μια ανάλυση και να μας δείξει πόσα από τα μηνύματα αυτά περιέχουν αρνητικές λέξεις-εκφράσεις . Δηλαδή πόσα από τα δεδομένα περιέχουν όχι και τόσο θετικά μηνύματα ή σκέψεις σχετικά με τον εμβολιασμό κατά του ιού Covid-19 . Η εφαρμογή αυτή θέλει να αποδείξει την αρνητικότητα-προβληματισμό-ανησυχία του κόσμου που επικρατεί έναντι του εμβολιασμού κατά του Covid-19 .

Η εφαρμογή αρχικά συλλέγει τα δεδομένα , μετά εμφανίζει κάποια στατιστικά στοιχεία και τέλος χωρίζει τα δεδομένα σε 2 ομάδες . Δεδομένα με θετικά μηνύματα ή σκέψεις σχετικά με τον εμβολιασμό κατά του ιού Covid-19 και δεδομένα με αρνητικά μηνύματα ή σκέψεις σχετικά με τον εμβολιασμό κατά του ιού Covid-19 .

ΕΝΟΤΗΤΑ 2: Αρχιτεκτονική Εφαρμογής

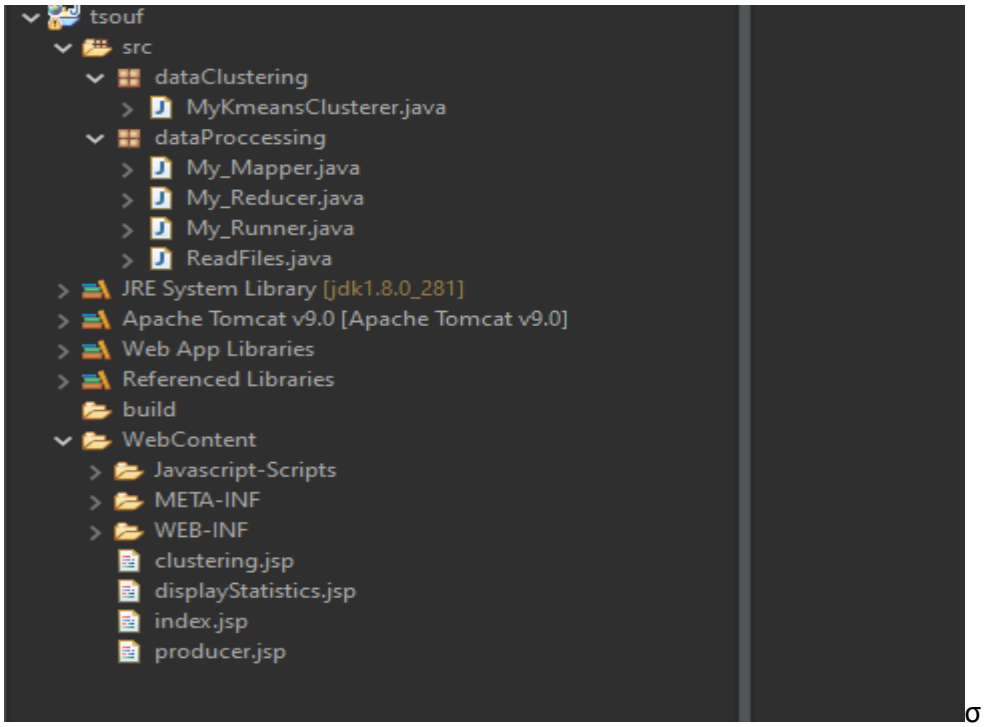
Η εφαρμογή έχει δομηθεί με jsp ,html , css ,bootStrap, javascript .Τα εργαλεία που χρησιμοποιήθηκαν για την συλλογή και επεξεργασία των δεδομένων είναι τα KAFKA , FLUME , HDFS, MapReduce , Mahout .

Πιο συγκεκριμένα έχουμε μία Java-Web Application εφαρμογή που όσον αφορά το front-end χρησιμοποιεί html , css ,bootStrap και javascript ενώ στο back-end χρησιμοποιείται η Java .

Η java στο backend χρησιμοποιεί τα εργαλεία KAFKA , FLUME , HDFS, MapReduce , Mahout έτσι ώστε να συλλέξει και να επεξεργαστεί τα live streaming δεδομένα .

Σε αρχικό στάδιο η εφαρμογή με τη βοήθεια του KAFKA και του FLUME συλλέγει τα δεδομένα και τα γράφει σε ένα φάκελο στο HDFS .Μετά εφαρμόζει ένα MapReduce job στα συλλεχθέντα δεδομένα με σκοπό να εμφανίσει κάποια στατιστικά .Μετά με την βοήθεια του Mahout η εφαρμογή χρησιμοποιεί τα συλλεχθέντα δεδομένα από το HDFS και τα προεπεξεργάζεται .Κατά την προεπεξεργασία των δεδομένων (tweets) διαχωρίζονται τα μηνύματα που έχουν αρνητική χροιά από τα υπόλοιπα. Τέλος εφαρμόζει στα προεπεξεργασμένα δεδομένα τον αλγόριθμο K-Means , με σκοπό να δούμε πόσα τελικά από τα 2000 tweets περιείχαν κάτι αρνητικό σχετικά με τον εμβολιασμό κατά του Covid-19 .

Ακολουθεί η δομή του Java-Web Application.



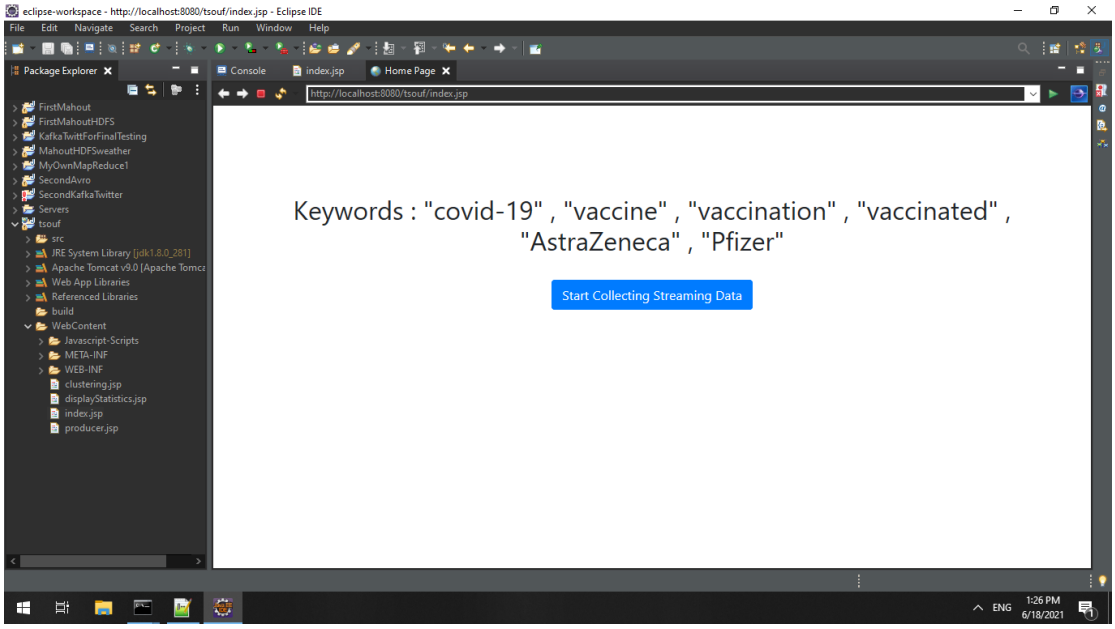
σ

ΕΝΟΤΗΤΑ 3: Εγχειρίδιο Χρήσης Εφαρμογής

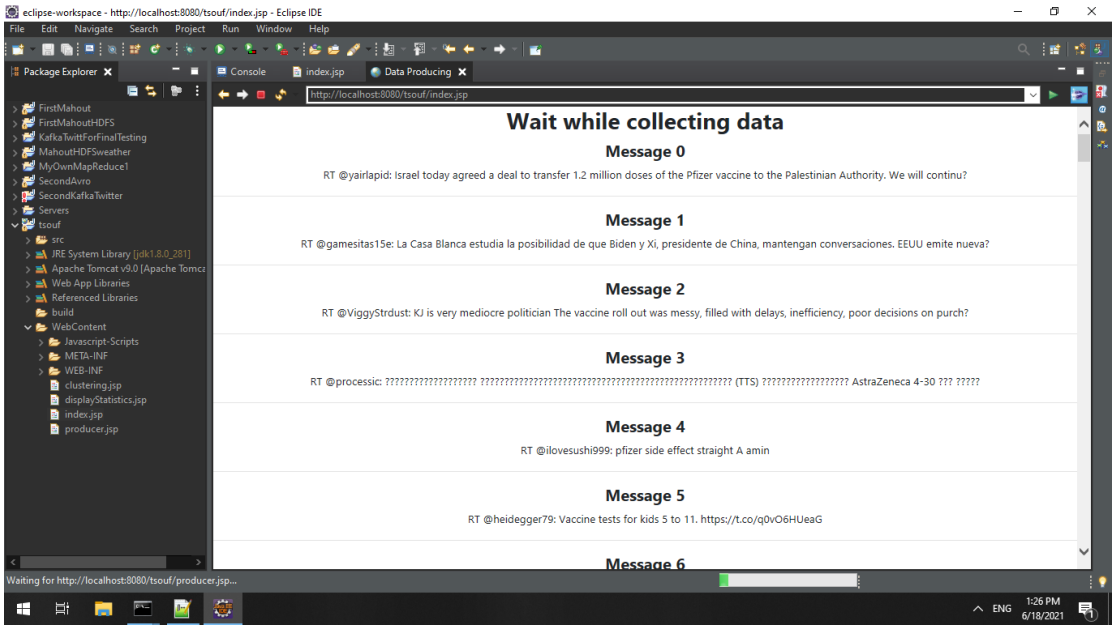
Για να τρέξει κάποιος την εφαρμογή πρέπει :

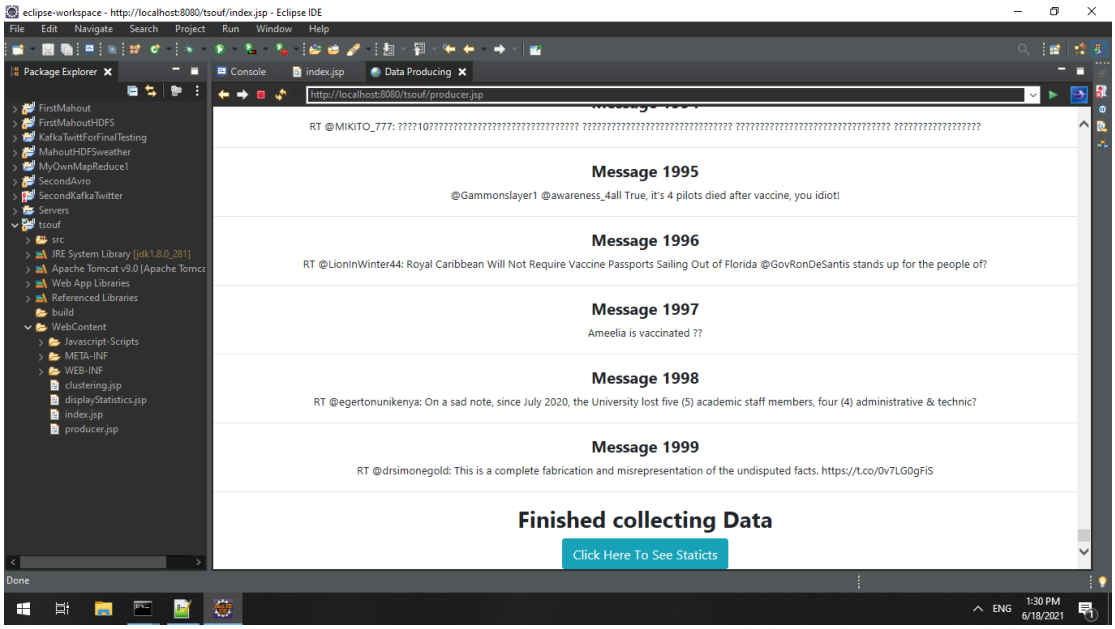
1) Να ανοίξει στο eclipse τον φάκελο με τα αρχεία της εφαρμογής μου (tsouf) .
2) Να έχει εγκαταστήσει σωστά τον Tomcat .
3) Να τρέχουν τα hdfs , yarn , zookeeper , kafka .
4) Να τρέχει το flume χρησιμοποιώντας το .conf αρχείο μου (twitter5.conf)
5) Να φτιάξει τοπικά στο hdfs του ένα φάκελο με το όνομα finalProject_file15
6) Να τρέξει την σελίδα index.jsp

Μόλις φορτωθεί η σελίδα θα δει το παρακάτω παράθυρο .Για να ξεκινήσει η διαδικασία αρκεί να πατήσει το μπλέ κουμπί .

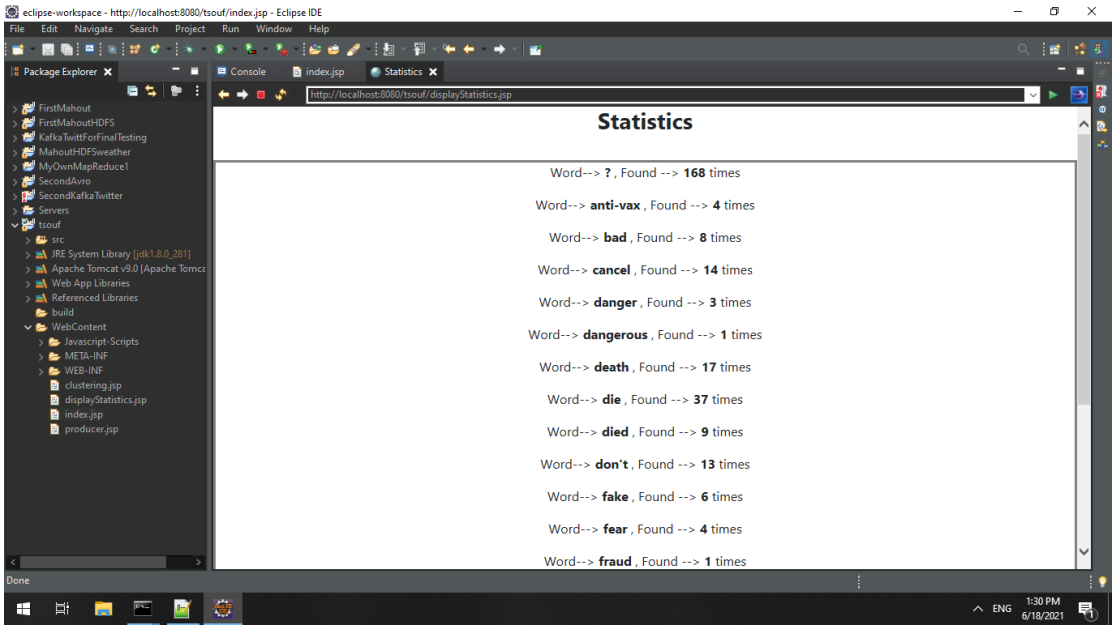


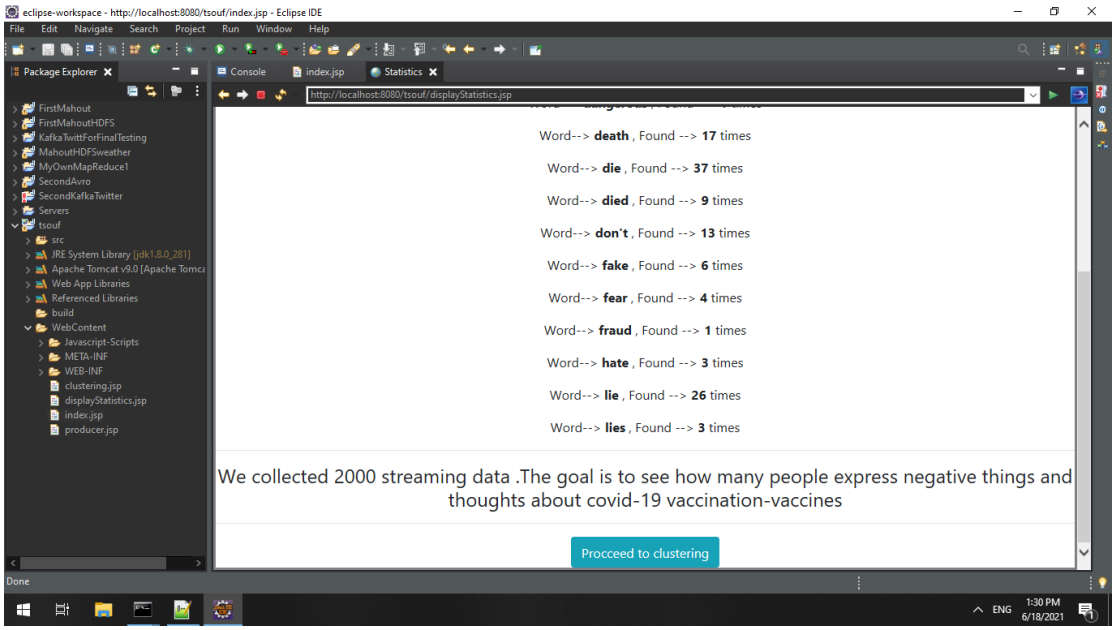
Στη συνέχεια περιμένει μέχρι να μαζευτούν 2000 streaming δεδομένα από το twitter με τις παραπάνω λέξεις κλειδιά.





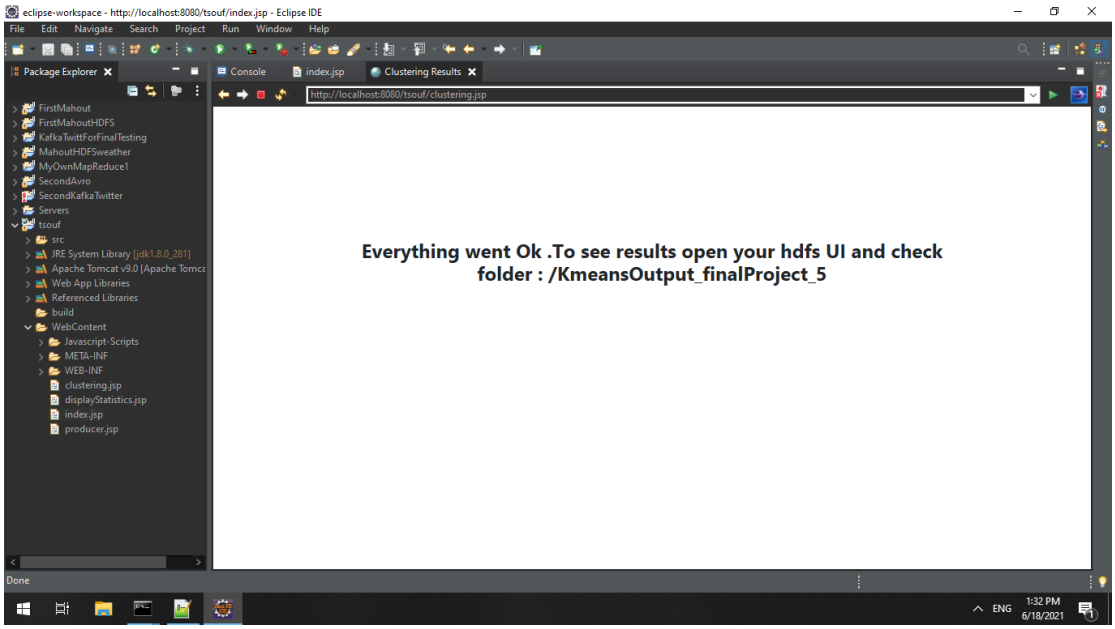
Τώρα πατάει το μπλέ κουμπί που φαίνεται από πάνω για να δει κάποια στατιστικά .Πρόκειται για την συχνότητα εμφάνισης κάποιων συγκεκριμένων λέξεων μέσα στα συλλεχθέντα tweets .



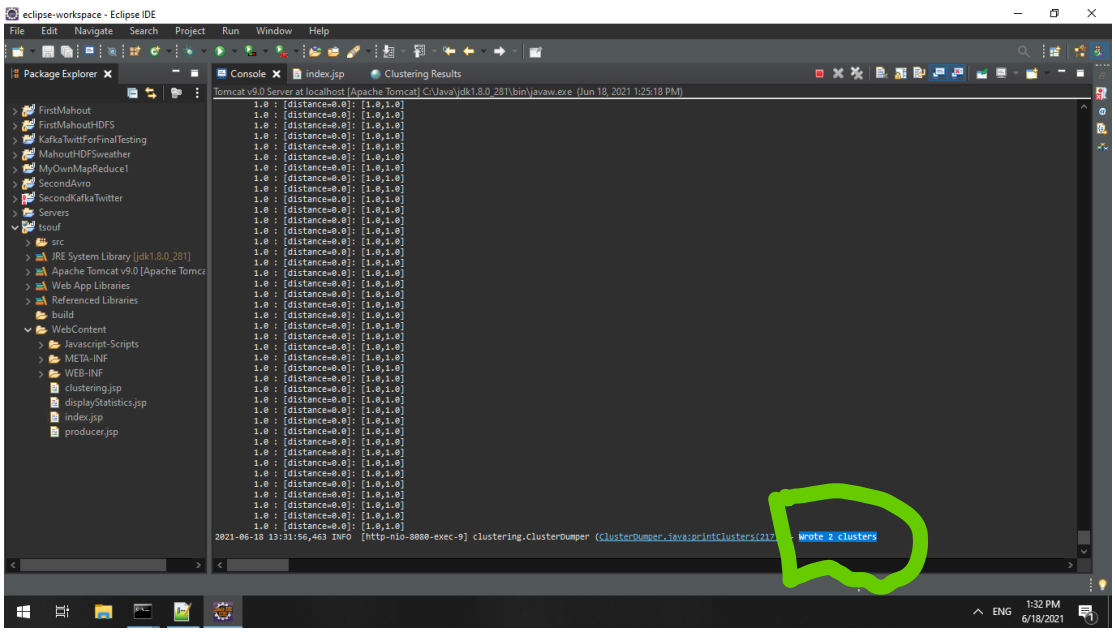


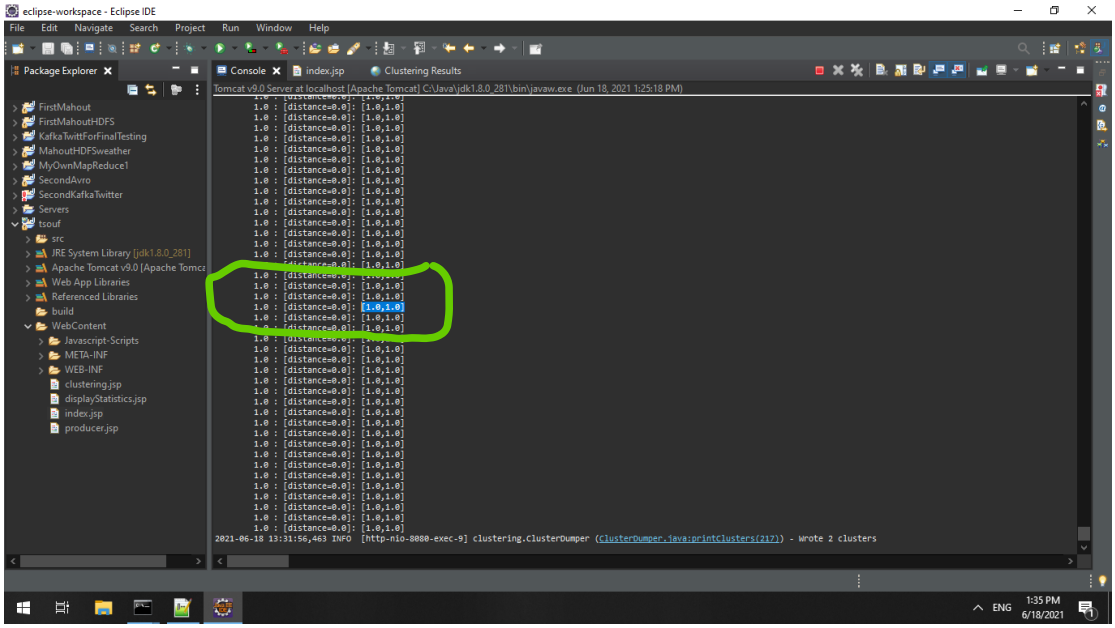
Τώρα θα πατήσει το κουμπί που φαίνεται παραπάνω έτσι ώστε να διαχωριστούν τα μηνύματα που έχουν αρνητική χροιά από τα υπόλοιπα .

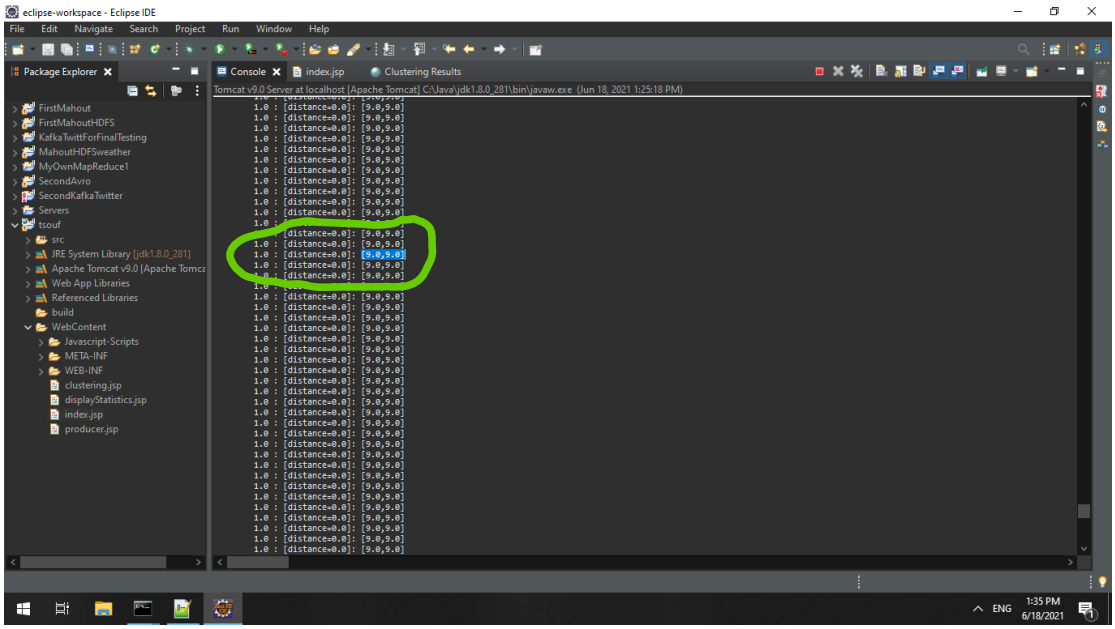
Τα αποτελέσματα θα αποθηκευτούν τοπικά στο hdfs του χρήστη , έτσι ώστε να μπορέσει να τα δεί και να επεξεργαστεί .Του εμφανίζεται και μήνυμα σωστής εκτέλεσης του προγράμματος .



Τα μηνύματα με συντεταγμένες 1,1 είναι αυτά που περιέχουν αρνητισμό-προβληματισμό .Τα υπόλοιπα έχουν συντεταγμένες 9,9



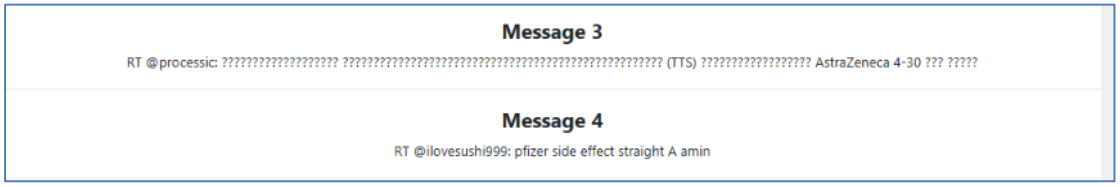

eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer
Console
index.jsp
Clustering Results
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Java\jdk1.8.0_281\bin\javaw.exe (Jun 18, 2021 1:25:18 PM)
1.0 : [distance=0.0]: [1.0,1.0]
2021-06-18 13:31:56,463 INFO [http-nio-8080-exec-9] clustering.ClusterDumper (ClusterDumper.java:printClusters(217)) - Wrote 2 clusters
FirstMahout
FirstMahoutHDFS
KafkaTwittForFinalTesting
MahoutHDFSweather
MyOwnMapReduce1
SecondAvro
SecondKafkaTwitter
Servers
tsouf
src
JRE System Library [jdk1.8.0_281]
Apache Tomcat v9.0 [Apache Tomcat
Web App Libraries
Referenced Libraries
build
WebContent
Javascript-Scripts
META-INF
WEB-INF
clustering.jsp
displayStatistics.jsp
index.jsp
producer.jsp
ENG
1:35 PM
6/18/2021


eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Package Explorer
Console
index.jsp
Clustering Results
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Java\jdk1.8.0_281\bin\javaw.exe (Jun 18, 2021 1:25:18 PM)
1.0 : [distance=0.0]: [9.0,9.0]
FirstMahout
FirstMahoutHDFS
KafkaTwittForFinalTesting
MahoutHDFSweather
MyOwnMapReduce1
SecondAvro
SecondKafkaTwitter
Servers
tsouf
src
JRE System Library [jdk1.8.0_281]
Apache Tomcat v9.0 [Apache Tomcat
Web App Libraries
Referenced Libraries
build
WebContent
Javascript-Scripts
META-INF
WEB-INF
clustering.jsp
displayStatistics.jsp
index.jsp
producer.jsp
ENG
1:35 PM
6/18/2021

ΕΝΟΤΗΤΑ 4: Αποτελέσματα Εφαρμογής

Οι λέξεις κλειδιά βάση των οποίων γίνεται η συλλογή δεδομένων , όπως αναφέρονται και στην αρχική σελίδα της εφαρμογής , είναι covid-19 , vaccine , vaccination , vaccinated , AstraZeneca , Pfizer .

Ακολουθούν παραδείγματα από τα συλλεχθέντα δεδομένα



Το configuration αρχείο που δημιουργήθηκε για το εργαλείο flume είναι το ακόλουθο :

```
KafkaAgent.sources = Kafka
KafkaAgent.sinks = HDFS
KafkaAgent.channels = MemChannel

KafkaAgent.sources.Kafka.type = org.apache.flume.source.kafka.KafkaSource
KafkaAgent.sources.Kafka.kafka.bootstrap.servers = localhost:9092
KafkaAgent.sources.Kafka.kafka.topics = finalProject_file15
KafkaAgent.sources.Kafka.kafka.consumer.group.id = flume
KafkaAgent.sources.Kafka.interceptors = i1
KafkaAgent.sources.Kafka.interceptors.i1.type= timestamp
KafkaAgent.sources.Kafka.kafka.consumer.timeout.ms = 100
KafkaAgent.sources.Kafka.batchSize = 100

KafkaAgent.sinks.HDFS.type = hdfs
KafkaAgent.sinks.HDFS.hdfs.path = hdfs://localhost:9000/finalProject_file15/
KafkaAgent.sinks.HDFS.hdfs.fileType = DataStream
KafkaAgent.sinks.HDFS.hdfs.writeFormat = Text
KafkaAgent.sinks.HDFS.hdfs.batchSize = 1000
KafkaAgent.sinks.HDFS.hdfs.rollSize = 0
KafkaAgent.sinks.HDFS.hdfs.rollCount = 1000


KafkaAgent.channels.MemChannel.type = memory
KafkaAgent.channels.MemChannel.capacity = 10000
KafkaAgent.channels.MemChannel.transactionCapacity = 1000

KafkaAgent.sources.Kafka.channels = MemChannel
KafkaAgent.sinks.HDFS.channel = MemChannel
```

Όπως φαίνεται ,και όπως ζητάει η άσκηση ,source είναι το kafka ενώ sink είναι το hdfs .

Το MapReduce job που έγινε στα συλλεχθέντα δεδομένα κάνει το εξής …

Ψάχνει σε κάθε tweet για κάποιες συγκεκριμένες λέξεις που υποδεικνύουν αρνητικότητα-προβληματισμό-φόβο και βρίσκει την συχνότητα εμφάνισης των λέξεων αυτών μέσα στα συλλεχθέντα δεδομένα .Το αποτέλεσμα το εμφανίζει η εφαρμογή στην διεπαφή της Statistics.jsp .

Κατά την προεπεργασία των δεδομένων , διαχωρίζονται όσα περιέχουν αρνητικότητα-προβληματισμό-φόβο από τα υπόλοιπα .

Τα συλλεχθέντα δεδομένα και τα αποτελέσματα της εφαρμογής αποδεικνύουν πως τα περισσότερα tweets που γίνονται και περιέχουν τις προαναφερθέντες λέξεις κλειδιά , έχουν ύφος αρνητικό και περιέχουν προβληματισμό ή και φόβο .